**AMENDMENTS TO THE CLAIMS:**

This listing of claims will replace all prior versions and listings of claims in the application:

1. (Currently Amended) A computer-implemented method for dynamic data type enrichment comprising the steps:

~~using at least one~~ loading an application program into memory, the application program comprising a variable that is defined as an instance of both a basic data type and a specific data type ~~in a predefined application program~~; ~~and~~

accessing metadata at runtime to map the variable to a definition of the specific data type; and

processing the variable consistently with the metadata definition of the specific data type

~~adding metadata to the at least one basic data type at runtime when the application program is executed to create a variable in the application program that comprises both the basic data type and the metadata~~.

2. (Currently Amended) The method of claim 1, wherein the application program uses an application programming interface for accessing the metadata ~~before adding the metadata to the basic data type~~.

3. (Previously Presented) The method of claim 2, wherein the application program calls through the application programming interface at least one metadata service that relates to the basic data type.

4. (Previously Presented) The method of claim 3, wherein the at least one metadata service copies the metadata to a metadata cache.

5. (Currently Amended) The method of claim 1, wherein the basic data type is defined in ~~a programming language~~ one or more of Visual Basic, Java, or C++ ~~used by the application program~~.

6. (Currently Amended) The method of claim [[5]] 1, wherein the metadata defines an allowed value range for the specific data type that is a subset of an allowed value range for the basic data type ~~is associated with a specific data type defined in a metadata store~~.

7. (Currently Amended) The method of claim [[6]] 1, wherein the metadata defines a text label for a user input field ~~application program provides a mapping between the specific data type and the basic data type~~.

8. (Currently Amended) The method of claim [[6]] 7, wherein the variable is set to a value entered into the user input field ~~application program uses a variable to map the specific data type to the basic data type~~.

9. (Previously Presented) The method of claim 6, wherein the metadata is stored in a private instance of the metadata store together with the application.

10. (Previously Presented) The method of claim 6, wherein the metadata is stored in a shared instance of the metadata store.

11. (Canceled)

12. (Currently Amended) A computer system comprising:

a memory storing an application program that ~~uses a~~ comprises a variable that is defined as an instance of both a basic data type and a specific data type; and

a processor executing instructions to:

access metadata at runtime to map the variable to a definition of the specific data type; and

process the variable consistently with the metadata definition of the specific data type

~~add metadata to the basic data type when executing the application program to create a variable in the application program that comprises both the basic data type and the metadata~~.

13. (Previously Presented) The computer system of claim 12 further comprising an application programming interface to access the metadata from the application program.

14. (Previously Presented) The computer system of claim 13, wherein the application programming interface provides at least one metadata service that relates to the basic data type used by the application program.

15. (Previously Presented) The computer system of claim 14, further comprising a metadata cache, the at least one metadata service copying the metadata to the metadata cache.

16. (Currently Amended) The computer system of claim 12, wherein the basic data type is defined in ~~a programming language~~ one or more of Visual Basic, Java, or C++ ~~used by the application program~~.

17. (Currently Amended) The computer system of claim 16, wherein the metadata defines an allowed value range for the specific data type that is a subset of an allowed value range for the basic data type ~~are associated with a specific data type defined in a metadata store~~.

18. (Currently Amended) The computer system of claim 17, wherein the metadata defines a text label for a user input field ~~application program provides a mapping between the specific data type and the basic data type~~.

19. (Currently Amended) The computer system of claim 18, wherein the variable is set to a value entered into the user input field ~~application program uses a variable to map the specific data type to the basic data type~~.

20. (Previously Presented) The computer system of claim 17, wherein the metadata is stored in a private instance of the metadata store together with the application.

21. (Previously Presented) The computer system of claim 17, wherein the metadata is stored in a shared instance of the metadata store.

22. (Withdrawn) A method for generating an application program (210) comprising the steps:

making available at least one metadata service (191) to be used in the application program (210) at design time for defining how the application program (210) can access metadata (150) at runtime; and

including a first implementation portion of the least one metadata service (191) in the IDE (800) that is unaffected by changes of a second implementation portion of the least one metadata service (191) in a metadata store (220).

23. (Withdrawn) An integrated development environment (IDE) (800) for generating an application program (210) by performing the steps of claim 22.

24. (Canceled)

25. (New) The method according to claim 1, wherein the variable is declared as an instance of both the basic data type and the specific data type at design time.

26. (New) A computer-implemented method for dynamic data type enrichment comprising:

loading an application program into memory, the application program comprising a variable that is defined as an instance of both a basic data type and a specific data type;

accessing metadata over a network at runtime using an application programming interface, the metadata mapping the variable to a definition of the specific data type that indicates a text label for an input field to the variable and indicating allowed value range for the variable; and

processing the variable consistently with the indicated allowed value range.

27. (New) A computer program product comprising instructions embodied on a memory of a computer system that cause at least one processor of the computer system to execute a method, the method comprising:

loading an application program into memory, the application program comprising a variable that is defined as an instance of both a basic data type and a specific data type;

accessing metadata at runtime to map the variable to a definition of the specific data type; and

processing the variable consistently with the metadata definition of the specific data type.

28. (New) The computer program product according to claim 27, wherein the metadata is stored on a hard disk, and the application program loaded into memory accesses the metadata by retrieving the metadata from the hard disk.